

也谈 PB 窗口自适应系统环境变化

摘要:用户需要在不同分辨率运行PowerBuilder应用, 并需调整窗口大小, 但此时窗口内控件的大小、相对位置不能随之调整。本文针对此问题, 提出了动态调整的通用方法。

关键词:PowerBuilder 动态调整 控件 类 对象

宋 铭 (海军装备论证研究中心系统所 100073)

拜读了《计算机系统应用》2000年第9期杜彩玲同志撰写的《PB 开发应用中窗口自适应系统环境变化》后, 很受启发, 感到作者思路独特, 并对各种环境的高度宽度调整值作了详细的阐述。

本文提出了对窗口内控件位置、尺寸动态调整的通用方法, 作为对杜彩玲同志文章的补充。

1 问题的提出

PowerBuilder的窗口对象是应用程序的主要图形用户界面(GUI), 窗口对象内可放置种类繁多的各种控件, 使得应用程序具有很强的表现力。

为了通用性, 设计应用程序时一般以640×480为标准分辨率, 由于种种原因, 用户经常使用其他的显示分辨率。当分辨率从640×480调整为800×600或1024×768时, 应用窗口相对变小, 用户为了充分利用屏幕空间而将窗口最大化时, 窗口内控件的尺寸、位置是固定的, 并且集中到窗口的左上角, 既不美观, 又浪费了宝贵的屏幕空间。另外, 缺省情况下窗口的Resizable选项是打开的, 用户可任意调整窗口的大小, 如控件不能动态调整, 则会出现控件遮盖现象。本文以PowerBuilder 6.5为例, 详细阐述实现控件动态调整尺寸、位置的方法。

2 预备知识

PowerBuilder是一种面向对象的开发工具, 打开Browser画板后选择System标签页面, 可以看到按字母顺序排列的所有对象种类。共有118个对象类, 单击鼠标右键, 从弹出式菜单中选择Show Hierarchy, 可看到所有对象种类的继承层次, 可以看出, 除了SystemFunctions类, 所有对象均由祖先类PowerObject继承而来。PowerObject有九种子类, 窗口内所有控件对象均继承自PowerObject\GraphicObject\WindowObject, 该类派生出两种子类:

DrawObject和DrogObject。DrawObject类是画图对象, 是不可拖放的, 属于该类控件有: 线段、椭圆、矩形和圆角矩形, 一般用于对窗口的美化修饰。除了这四种控件, 其余二十四种控件均属于DrogObject类, 是可拖放对象。

3 实现原理

PowerBuilder的窗口对象有一个非常重要的属性——Control [] 数组, 它存储了窗口内放置的所有控件(运行时动态建立的控件除外), 控件的数量即为数组维数, 数组类型为WindowObject。其中DrogObject有X、Y、Width、Height属性; DrawObject类中的椭圆、矩形和圆角矩形控件也有X、Y、Width、Height属性, 但DrawObject类中的线段控件中只有StartX、StartY、EndX、EndY属性, 所以DrawObject类中没有这些属性。

利用窗口事件Resize, 该事件有两个整型参数NewWidth、NewHeight, 代表改变尺寸后的窗口宽度、高度, 与窗口原高度、宽度比较, 得到两个比例x_proportion、y_proportion, 分别利用这两个比例, 计算出控件的X、Y、Width、Height属性(对于线段控件计算StartX、StartY、EndX、EndY属性), 即实现了窗口内控件尺寸的动态调整。对DragObject类可统一编程, 对DrawObject类中的线段、椭圆、矩形和圆角矩形控件则需分别编程。

4 解决方案

在窗口画板建立新窗口w_ancestor, 定义以下实例变量:

```
Private: // 为了创建高度封装的对象, 禁止该类外部的对象以及该类的后代
```

```
// 对以下窗口实例变量的访问
```

```
Int i,j,k,l,m,n // 循环、可拖动控件、线段、椭圆、矩形、圆角矩形计数器
```

```

DragObject a [] //可拖动控件数组
Line b [] //线段数组
Oval c [] //椭圆数组
Rectangle d [] //矩形数组
RoundRectangle e [] //圆角矩形数组
Real ww,hh //窗口上次的宽度、高度
在其 Open 事件中编写以下脚本：
//将 Control [] 数组分门别类放入 a [], b [], c [],
d [], e [] 中
FOR i=1 TO UpperBound(Control)
CHOOSE CASE Control [i] .TypeOf()
CASE Line! //线段
    j=j+1; b [j] =Control [i]
CASE Oval! //椭圆
    k=k+1; c [k] =Control [i]
CASE Rectangle! //矩形
    l=l+1; d [l] =Control [i]
CASE RoundRectangle! //圆角矩形
    m=m+1; e [m] =Control [i]
CASE ELSE //可拖动控件
    n=n+1; a [n] =Control [i]
END CHOOSE
NEXT
//将窗口初始宽度、高度存入变量 ww、hh
ww=this.Width
hh=this.Height
在 Resize 事件中编写以下脚本：
Real x_proportion,y_proportion //x、y 方向调整比例
//防止被 0 除
IF NewWidth=0 or NewHeight=0 THEN
    return
END IF
//计算调整比例
x_proportion=NewWidth /ww
y_proportion=NewHeight/hh
//调整可拖动控件数组
FOR i=1 TO n
    a [i] .X = a [i] .X * x_proportion
    a [i] .Y = a [i] .Y * y_proportion
    a [i] .Width = a [i] .Width * x_proportion
    a [i] .Height = a [i] .Height * y_proportion
NEXT

```

```

//调整线段控件数组
FOR i=1 TO k
    b [i] .BeginX = b [i] .BeginX * x_proportion
    b [i] .BeginY = b [i] .BeginY * y_proportion
    b [i] .EndX = b [i] .EndX * x_proportion
    b [i] .EndY = b [i] .EndY * y_proportion
NEXT
//调整椭圆控件数组
FOR i=1 TO j
    c [i] .X = c [i] .X * x_proportion
    c [i] .Y = c [i] .Y * y_proportion
    c [i] .Width = c [i] .Width * x_proportion
    c [i] .Height = c [i] .Height * y_proportion
NEXT
//调整矩形控件数组
FOR i=1 TO l
    d [i] .X = d [i] .X * x_proportion
    d [i] .Y = d [i] .Y * y_proportion
    d [i] .Width = d [i] .Width * x_proportion
    d [i] .Height = d [i] .Height * y_proportion
NEXT
//调整圆角矩形控件数组
FOR i=1 TO m
    e [i] .X = e [i] .X * x_proportion
    e [i] .Y = e [i] .Y * y_proportion
    e [i] .Width = e [i] .Width * x_proportion
    e [i] .Height = e [i] .Height * y_proportion
NEXT
//将窗口现宽度、高度存入变量 ww、hh，为下一次
调整做好准备
ww=NewWidth
hh=NewHeight

```

5 其他说明

如果能够确定整个应用不使用线段、椭圆、矩形和圆角矩形控件,则以上关于这四种控件的代码及变量声明均可删除,这样可以在很大程度上简化程序。由于构成一个完整应用程序所需建立的窗口众多,单独给每个窗口编程是件很繁琐的工作,利用 PowerBuilder 面向对象的特征,将本文例子中的 w_ancestor 作为应用的祖先窗口,其他窗口均继承该窗口或该窗口的后代窗口,则解决了在整个应用程序内动态调整窗口内控件尺寸、位置的问题。■